

**Amendments to the Claims:**

This listing of claims will replace all prior versions and listings of claims in the application.

**Listing of Claims:**

1. (cancelled)
2. (cancelled)
3. (cancelled)
4. (cancelled)
5. (cancelled)
6. (new) A process of detecting security vulnerabilities present in a target Web site, comprising:

establishing an Internet connection with the target Web site;

retrieving a default Web page for the target Web site;

parsing through the default Web page to identify any linked-to Web pages or objects which are included in the default Web page;

automatically passing an authorized username and password to the target Web site, if required to gain access to the target Web site;

scanning the target Web site for at least one known exploit in order to identify security vulnerabilities;

applying at least one predetermined hack method to the target Web site in order to identify security vulnerabilities; and

outputting the security vulnerabilities.

7. (new) The method of claim 6, further comprising scanning at least one of the security vulnerabilities for at least one known exploit in order to identify further security vulnerabilities.

8. (new) The method of claim 6, further comprising parsing through the linked-to Web pages to identify any further-linked-to Web pages or objects which are included in the linked-to Web pages.

9. (new) The method of claim 8, further comprising parsing through the default Web page to identify any hidden Web pages or objects which are included in the hidden Web pages.

10. (new) The method of claim 9, further comprising parsing through the hidden Web-pages to identify any further-hidden Web pages or objects which are included in the further-hidden Web pages.

11. (new) The method of claim 10, further comprising:  
comparing each hidden Web page and each further-hidden Web page to each linked-to Web page and each further-linked-to Web page; and  
identifying each hidden Web page and each further-hidden Web page that is different from the linked-to Web pages and the further-linked to Web pages.

12. (new) The method of claim 8, wherein the parsing through the default Web page and the parsing through the linked-to Web pages include performing a keyword search in order to detect at least one point of interest.

13. (new) The method of claim 12, wherein the at least one point of interest is selected from the group consisting of an administration Web page and a directory list tag.

14. (new) The method of claim 12, wherein the applying at least one predetermined hack method includes attempting to access unauthorized files located outside the target Web site's root directory.

15. (new) The method of claim 8, wherein the applying at least one predetermined hack method includes attempting to access unauthorized files located outside the target Web site's root directory.

16. (new) The method of claim 15, wherein the scanning the target Web site for at least one known exploit includes checking for at least one common filename.

17. (new) The method of claim 16, wherein the at least one common filename is selected from the group consisting of "msadcs.dll" and "WS\_FTP.LOG."

18. (new) The method of claim 8, wherein the applying at least one predetermined hack method includes automatically passing multiple usernames and passwords to the target Web site if a login Web page is encountered.

19. (new) A process of detecting security vulnerabilities present in a target Web site, comprising:

establishing an Internet connection with the target Web site;

retrieving a default Web page for the target Web site;

parsing through the default Web page to identify any linked-to Web pages or objects which are included in the default Web page, wherein the parsing includes performing a keyword search in order to detect at least one point of interest;

scanning the target Web site for at least one known exploit in order to identify security vulnerabilities;

applying at least one predetermined hack method to the target Web site in order to identify security vulnerabilities; and  
prioritizing the security vulnerabilities.

20. (new) The method of claim 19, further comprising parsing through the default Web page to identify any hidden Web pages or objects which are included in the hidden Web pages.

21. (new) The method of claim 20, further comprising parsing through the hidden Web-pages to identify any further-hidden Web pages or objects which are included in the further-hidden Web pages.

22. (new) The method of claim 21, further comprising:  
comparing each hidden Web page and each further-hidden Web page to each linked-to Web page and each further-linked-to Web page; and  
identifying each hidden Web page and each further-hidden Web page that is different from the linked-to Web pages and the further-linked to Web pages.

23. (new) The method of claim 19, further comprising parsing through the linked-to Web pages to identify any further-linked-to Web pages or objects which are included in the linked-to Web pages.

24. (new) The method of claim 23, wherein the at least one point of interest is selected from the group consisting of an administration Web page and a directory list tag.

25. (new) The method of claim 23, further comprising scanning at least one of the security vulnerabilities for at least one known exploit in order to identify further security vulnerabilities.

26. (new) The method of claim 23, wherein the applying at least one predetermined hack method includes attempting to access unauthorized files located outside the target Web site's root directory.

27. (new) The method of claim 23, further comprising automatically passing an authorized username and password to the target Web site, if required to gain access to the target Web site.

28. (new) The method of claim 23, wherein the applying at least one predetermined hack method includes automatically passing multiple usernames and passwords to the target Web site if a login Web page is encountered.

29. (new) The method of claim 23, wherein the applying at least one predetermined hack method includes passing invalid data to a data entry field of the target Web site and evaluating the result.

30. (new) The method of claim 29, further comprising:  
recording the invalid data which produces a security vulnerability; and  
passing the recorded invalid data to at least one other data entry field of the target Web site.

31. (new) A process of detecting security vulnerabilities present in a target Web site, comprising:

establishing an Internet connection with the target Web site;  
retrieving a default Web page for the target Web site;  
parsing through the default Web page to identify any linked-to Web pages or objects which are included in the default Web page;

scanning the target Web site for at least one known exploit in order to identify security vulnerabilities;

applying at least one predetermined hack method to the target Web site in order to identify security vulnerabilities, wherein the applying at least one predetermined hack method includes attempting to access unauthorized files located outside the target Web site's root directory; and

outputting the security vulnerabilities.

32. (new) The method of claim 31, further comprising scanning at least one of the security vulnerabilities for at least one known exploit in order to identify further security vulnerabilities.

33. (new) The method of claim 31, further comprising parsing through the linked-to Web pages to identify any further-linked-to Web pages or objects which are included in the linked-to Web pages.

34. (new) The method of claim 33, further comprising parsing through the default Web page to identify any hidden Web pages or objects which are included in the hidden Web pages.

35. (new) The method of claim 34, further comprising parsing through the hidden Web-pages to identify any further-hidden Web pages or objects which are included in the further-hidden Web pages.

36. (new) The method of claim 35, further comprising:  
comparing each hidden Web page and each further-hidden Web page to each linked-to Web page and each further-linked-to Web page; and

identifying each hidden Web page and each further-hidden Web page that is different from the linked-to Web pages and the further-linked to Web pages.

37. (new) The method of claim 33, wherein the parsing through the default Web page and the parsing through the linked-to Web pages include performing a keyword search in order to detect at least one point of interest.

38. (new) The method of claim 37, wherein the at least one point of interest is selected from the group consisting of an administration Web page and a directory list tag.

39. (new) The method of claim 38, further comprising automatically passing an authorized username and password to the target Web site, if required to gain access to the target Web site.

40. (new) The method of claim 33, wherein the scanning the target Web site for at least one known exploit includes checking for at least one common filename.

41. (new) The method of claim 40, wherein the at least one common filename is selected from the group consisting of “msadcs.dll” and “WS\_FTP.LOG.”

42. (new) A system for detecting security vulnerabilities present in a target Web site, comprising:

memory for storing:

a Web page database;

at least one exploit; and

a security vulnerability database; and

a processor connected to the memory and being configured to:

establish an Internet connection with the target Web site;

retrieve a default Web page for the target Web site;  
parse through the default Web page to identify any linked-to Web pages or objects which are included in the default Web page;  
automatically pass an authorized username and password to the target Web site, if required to gain access to the target Web site;  
scan the target Web site for at least one known exploit in order to identify security vulnerabilities;  
apply at least one predetermined hack method to the target Web site in order to identify security vulnerabilities; and  
prioritize the security vulnerabilities.

43. (new) The system of claim 42, wherein the processor is further configured to parse through the linked-to Web pages to identify any further-linked-to Web pages or objects which are included in the linked-to Web pages.

44. (new) The system of claim 43, wherein the processor is further configured to scan at least one of the security vulnerabilities for at least one known exploit in order to identify further security vulnerabilities.

45. (new) The system of claim 44, wherein the processor is further configured to parse through the default Web page to identify any hidden Web pages or objects which are included in the hidden Web pages.

46. (new) The system of claim 45, wherein the processor is further configured to parse through the hidden Web-pages to identify any further-hidden Web pages or objects which are included in the further-hidden Web pages.

47. (new) The system of claim 46, wherein the processor is further configured to:  
compare each hidden Web page and each further-hidden Web page to each  
linked-to Web page and each further-linked-to Web page; and

identify each hidden Web page and each further-hidden Web page that is  
different from the linked-to Web pages and the further-linked to Web pages.

48. (new) The system of claim 43, wherein the applying at least one  
predetermined hack method includes attempting to access unauthorized files located outside  
the target Web site's root directory.

49. (new) The system of claim 48, wherein the parsing through the default Web  
page and the parsing through the linked-to Web pages include performing a keyword search  
in order to detect at least one point of interest.

50. (new) The system of claim 49, wherein the at least one point of interest is  
selected from the group consisting of an administration Web page and a directory list tag.

51. (new) The system of claim 43, wherein the applying at least one  
predetermined hack method includes automatically passing multiple usernames and  
passwords to the target Web site if a login Web page is encountered.

52. (new) A system for detecting security vulnerabilities present in a target Web  
site, comprising:

memory for storing:  
a Web page database;  
at least one exploit; and  
a security vulnerability database; and

a processor connected to the memory and being configured to:

establish an Internet connection with the target Web site;

retrieve a default Web page for the target Web site;

parse through the default Web page to identify any linked-to Web pages or objects which are included in the default Web page, wherein the parsing includes performing a keyword search in order to detect at least one point of interest;

scan the target Web site for at least one known exploit in order to identify security vulnerabilities;

apply at least one predetermined hack method to the target Web site in order to identify security vulnerabilities; and

output the security vulnerabilities.

53. (new) The system of claim 52, wherein the processor is further configured to scan at least one of the security vulnerabilities for at least one known exploit in order to identify further security vulnerabilities.

54. (new) The system of claim 52, wherein the processor is further configured to parse through the linked-to Web pages to identify any further-linked-to Web pages or objects which are included in the linked-to Web pages.

55. (new) The system of claim 54, wherein the at least one point of interest is selected from the group consisting of an administration Web page and a directory list tag.

56. (new) The system of claim 54, wherein the processor is further configured to parse through the default Web page to identify any hidden Web pages or objects which are included in the hidden Web pages.

57. (new) The system of claim 56, wherein the processor is further configured to parse through the hidden Web-pages to identify any further-hidden Web pages or objects which are included in the further-hidden Web pages.

58. (new) The system of claim 57, wherein the processor is further configured to:  
compare each hidden Web page and each further-hidden Web page to each linked-to Web page and each further-linked-to Web page; and  
identify each hidden Web page and each further-hidden Web page that is different from the linked-to Web pages and the further-linked to Web pages.

59. (new) The system of claim 54, wherein the processor is further configured to automatically pass an authorized username and password to the target Web site, if required to gain access to the target Web site.

60. (new) The system of claim 59, wherein the applying at least one predetermined hack method includes attempting to access unauthorized files located outside the target Web site's root directory.

61. (new) The system of claim 54, wherein the applying at least one predetermined hack method includes passing invalid data to a data entry field of the target Web site and evaluating the result.

62. (new) The system of claim 61, wherein the processor is further configured to:  
record the invalid data which produces a security vulnerability; and  
pass the recorded invalid data to at least one other data entry field of the target Web site.

63. (new) A system for detecting security vulnerabilities present in a target Web site, comprising:

memory for storing:

a Web page database;

at least one exploit; and

a security vulnerability database; and

a processor connected to the memory and being configured to:

establish an Internet connection with the target Web site;

retrieve a default Web page for the target Web site;

parse through the default Web page to identify any linked-to Web pages or objects which are included in the default Web page;

scan the target Web site for at least one known exploit in order to identify security vulnerabilities;

apply at least one predetermined hack method to the target Web site in order to identify security vulnerabilities, wherein the applying at least one predetermined hack method includes attempting to access unauthorized files located outside the target Web site's root directory; and

output the security vulnerabilities.

64. (new) The system of claim 63, wherein the processor is further configured to parse through the linked-to Web pages to identify any further-linked-to Web pages or objects which are included in the linked-to Web pages.

65. (new) The system of claim 64, wherein the processor is further configured to parse through the default Web page to identify any hidden Web pages or objects which are included in the hidden Web pages.

66. (new) The system of claim 65, wherein the processor is further configured to parse through the hidden Web-pages to identify any further-hidden Web pages or objects which are included in the further-hidden Web pages.

67. (new) The system of claim 66, wherein the processor is further configured to:  
compare each hidden Web page and each further-hidden Web page to each linked-to Web page and each further-linked-to Web page; and  
identify each hidden Web page and each further-hidden Web page that is different from the linked-to Web pages and the further-linked to Web pages.

68. (new) The system of claim 67, wherein the parsing through the default Web page and the parsing through the linked-to Web pages include performing a keyword search in order to detect at least one point of interest.

69. (new) The system of claim 68, wherein the at least one point of interest is selected from the group consisting of an administration Web page and a directory list tag.

70. (new) The system of claim 64, wherein the processor is further configured to automatically pass an authorized username and password to the target Web site, if required to gain access to the target Web site.

71. (new) The system of claim 70, wherein the processor is further configured to scan at least one of the security vulnerabilities for at least one known exploit in order to identify further security vulnerabilities.

72. (new) A method for detecting security vulnerabilities in a web application executing on a web server or web application server, comprising:

actuating the application in order to discover pre-defined elements of the application's interface with external clients;

generating client requests having unauthorized values for said elements in order to generate exploits unique to the application;

attacking the application using the exploits; and

evaluating the results of the attack;

wherein actuating the application includes:

sending an authorized client request in order to receive a server response;

parsing the response in order to discover links encapsulated therein; and

actuating discovered links in accordance with authorized client functionality in order to generate additional authorized client requests.

73. (new) The method according to claim 72, further including comparing discovered links to a filter and not generating authorized client requests for links matching the filter.

74. (new) The method according to claim 72, wherein said application interface elements are discovered by parsing at least one of the authorized client requests and server responses resulting therefrom.

75. (new) The method according to claim 74, further including analyzing the server responses in order to extract attributes of said application interface elements.

76. (new) A method for detecting security vulnerabilities in a hypertext-based web application installed on a web server or web application server, comprising:

traversing the application in order to discover and actuate links therein;

analyzing messages that flow or would flow between an authorized client and the web server in order to discover elements of the application's interface with external clients and attributes of said elements;

generating unauthorized client requests in which said elements are mutated;

sending the mutated client requests to the server; and

receiving server responses to the unauthorized client requests and evaluating the results thereof;

wherein traversing the application includes:

sending an authorized client request in order to receive a server response;

parsing the response in order to discover links encapsulated therein; and

actuating discovered links in accordance with authorized client functionality in order to receive authorized server responses from which additional authorized client requests can be generated.

77. (new) The method according to claim 76, further including comparing discovered links to a filter and not generating authorized client requests for links matching the filter.

78. (new) The method according to claim 76, wherein, in the event the authorized client request requires user-interactive parameters, supplying pre-configured values therefor.

79. (new) The method according to claim 76, wherein said application interface elements are discovered by parsing at least one of the authorized client requests and server responses resulting therefrom.

80. (new) The method according to claim 79, further including analyzing the server responses in order to extract attributes of said application interface elements.

81. (new) A scanner system, provided on a computer, for detecting security vulnerabilities in a HTML-based web application installed on a web server or web application server, the scanner system comprising:

a crawling engine for traversing the application in order to discover and actuate links therein;

an analysis engine for analyzing messages that flow or would flow between an authorized client and the web server in order to discover elements of the application's interface with external clients and attributes of said elements and for generating unauthorized client requests in which said elements are mutated; and

an attack engine for sending the mutated client requests to the server; receiving server responses to the unauthorized client requests and evaluating the results thereof.

82. (new) The scanner system according to claim 81, wherein the crawling engine:  
sends an authorized client request in order to receive a server response;  
invokes the parsing engine to parse the response in order to discover links encapsulated therein; and

actuates discovered links in accordance with authorized client functionality in order to receive authorized server responses from which additional authorized client requests can be generated.

83. (new) The scanner system according to claim 82, wherein the crawling engine compares discovered links to a filter and does not generate authorized client requests for filtered links.

84. (new) The scanner system according to claim 82, wherein, in the event the authorized client request requires user-interactive parameters, the crawling engine supplies pre-configured values therefor.

85. (new) A crawling engine, provided on a computer, for automatically traversing a hypertext-based web site, comprising:

means for sending a client request in order to receive a server response;

means for parsing the response in order to discover links encapsulated therein;

means for actuating one or more discovered links in accordance with authorized client functionality in order to receive one or more server responses from which one or more additional client requests are generated; and

means for automatically supplying values to user-interactive parameters in the additional client requests, if required.

86. (new) The engine according to claim 85, further including means for comparing discovered links to a filter and not generating client requests for filtered links.

87. (new) A computer program product comprising a computer readable medium having computer readable code embodied therein, the computer readable code, when

executed, causing a computer to implement a method for detecting security vulnerabilities in a web application executing on a web server or web application server, comprising:

actuating the application in order to discover pre-defined elements of the application's interface with external clients;

generating client requests having unauthorized values for said elements in order to generate exploits unique to the application;

attacking the application using the exploits; and

evaluating the results of the attack.

88. (new) The computer program product according to claim 87, wherein an application interface element is a path parameter.

89. (new) The computer program product according to claim 87, wherein an application interface element is a data parameter.

90. (new) The computer program product according to claim 87, wherein, in the implemented method, actuating the application includes:

sending an authorized client request in order to receive a server response;

parsing the response in order to discover links encapsulated therein; and

actuating discovered links in accordance with authorized client functionality in order to generate additional authorized client requests.

91. (new) The computer program product according to claim 90, wherein the implemented method includes comparing discovered links to a filter and not generating authorized client requests for links matching the filter.

92. (new) The computer program product according to claim 90, wherein, in the implemented method, said application interface elements are discovered by parsing at least one of the authorized client requests and server responses resulting therefrom.

93. (new) The computer program product according to claim 92, wherein the implemented method includes analyzing the server responses in order to extract attributes of said application interface elements.

94. (new) A computer program product comprising a computer readable medium having computer readable code embodied therein, the computer readable code, when executed, causing a computer to implement a method for detecting security vulnerabilities in a hypertext-based web application installed on a web server or web application server, comprising:

traversing the application in order to discover and actuate links therein;  
analyzing messages that flow or would flow between an authorized client and the web server in order to discover elements of the application's interface with external clients and attributes of said elements;

generating unauthorized client requests in which said elements are mutated;  
sending the mutated client requests to the server; and  
receiving server responses to the unauthorized client requests and evaluating the results thereof.

95. (new) The computer program product according to claim 94, wherein an application interface element is a path parameter.

96. (new) The computer program product according to claim 94, wherein an application interface element is a data parameter.

97. (new) The computer program product according to claim 94, wherein an application interface element is a cookie.

98. (new) The computer program product according to claim 94, wherein, in the implemented method, traversing the application includes:

sending an authorized client request in order to receive a server response;

parsing the response in order to discover links encapsulated therein; and

actuating discovered links in accordance with authorized client functionality in order to receive authorized server responses from which additional authorized client requests can be generated.

99. (new) The computer program product according to claim 98, wherein the implemented method includes comparing discovered links to a filter and not generating authorized client requests for links matching the filter.

100. (new) The computer program product according to claim 98, wherein, in the implemented method, in the event the authorized client request requires user-interactive parameters, supplying pre-configured values therefor.

101. (new) The computer program product according to claim 98, wherein, in the implemented method, said application interface elements are discovered by parsing at least one of the authorized client requests and server responses resulting therefrom.